CS100M Section Exercise 5

1. Write a function **aprime(m)** that has an input parameter **m**. Function **aprime(m)** returns 1 if **m** is prime, and 0 otherwise. Remember to write a concise comment to describe the function, including its parameters under the function header.

2. A *twin prime* is a pair of primes such that if p is a prime, $p+2$ is also a prime. The larger prime in the pair is called the *big* prime, while the smaller prime is called the *little* prime. For example, in the twin prime pair (3,5), 5 is the big prime while 3 is the little prime. Write a function **lastTwinPair(n)** that will, given a number **n** greater than or equal to 5, return the last (largest) twin prime pair smaller than or equal to **n**. *Use function* **aprime** *from the previous question!* This function returns two values. Call them **littlep** and **bigp**.

3. [Modified from FA07 Prelim 1] The value of $\pi/8$ can be approximated by the series

$$1 + \frac{1}{3^2} + \frac{1}{5^2} + \frac{1}{7^2} + \cdots$$

Let the sum of the first n terms be T_n. As n increases one expects the ratio $\frac{T_n}{T_{n+1}}$ to approach 1. Write a script to find the smallest n such that the ratio $\frac{T_n}{T_{n+1}} > .9999$. Display n and T_n. *Do not* use arrays.

Hint: You need to keep track of a current sum and the next sum in the loop.

4. For each of the following sub-problems, complete the program so that it produces the desired result. You should not modify the given code in any way—only fill in the blanks that are provided. In every case you will need to use `for`-loops with an "increment" that is not one.

(a) The following program reads an integer k, and outputs all the multiples of k up to 1000.

```
k = input('Please enter a positive integer smaller than 1000:  ');
for j = ______________________
   fprintf('%d ', j);
end
fprintf('\n');
```

(b) The following program reads in a real number x and an integer N, and computes the sum $\sum_{k=0}^{N} \frac{(-1)^k x^{2k}}{(2k)!}$ to the first N terms. (This sum converges to $\cos(x)$ as $N \to \infty$.)

```
x = input('Please input a real number between 0 and pi/2:  ');
N = input('Please input a positive integer:  ');
sum = 0;
for j = ________________________
   sum = sum + (-1)^(j/2) * x^j / factorial(j);
end
fprintf('The sum of the first %d terms is %12.8f\n', N, sum);
```

(c) The following does the same thing as in part ***(b)***, but this time we are not allowed to use exponentiation and the `factorial` function, and must compute these explicitly.

```
x = input('Please input a real number between 0 and pi/2:  ');
N = input('Please input a positive integer:  ');
sum   = 1;    % Explicitly assign the first term (when j=0)
sign  = 1;    % The sign of a term, either 1 or -1
jfact = 1;    % Current value of j!
xtoj  = 1;    % Current value of x^j
for j = _________________________
   sign  = _________________________________;
   jfact = _________________________________;
   xtoj  = _________________________________;
   sum   = ___________________________________________;
end
fprintf('The sum of the first %d terms is %12.8f\n', N, sum);
```